

JAVA.LANG.STRICTMATH CLASS

http://www.tutorialspoint.com/java/lang/java_lang_strictmath.htm

Copyright © tutorialspoint.com

Introduction

The **java.lang.StrictMath** class contains methods for performing basic numeric operations such as the elementary exponential, logarithm, square root, and trigonometric functions.

Class declaration

Following is the declaration for **java.lang.StrictMath** class:

```
public final class StrictMath
    extends Object
```

Field

Following are the fields for **java.lang.StrictMath** class:

- **static double E** -- This is the double value that is closer than any other to e , the base of the natural logarithms.
- **static double PI** -- This is the double value that is closer than any other to π , the ratio of the circumference of a circle to its diameter.

Class methods

S.N.	Method & Description
1	<u>static double abs(double a)</u> This method returns the absolute value of a double value.
2	<u>static float abs(float a)</u> This method returns the absolute value of a float value.
3	<u>static int abs(int a)</u> This method returns the absolute value of an int value.
4	<u>static long abs(long a)</u> This method returns the absolute value of a long value.
5	<u>static double acos(double a)</u> This method returns the arc cosine of a value; the returned angle is in the range 0.0 through π .
6	<u>static double asin(double a)</u> This method returns the arc sine of a value; the returned angle is in the range $-\pi/2$ through $\pi/2$.
7	<u>static double atan(double a)</u> This method returns the arc tangent of a value; the returned angle is in the range $-\pi/2$ through $\pi/2$.
8	<u>static double atan2(double y, double x)</u> This method returns the angle theta from the conversion of rectangular coordinates (x, y) to polar coordinates (r, theta).

9	<u>static double cbrt(double a)</u> This method returns the cube root of a double value.
10	<u>static double ceil(double a)</u> This method returns the smallest (closest to negative infinity) double value that is greater than or equal to the argument and is equal to a mathematical integer.
11	<u>static double copySign(double magnitude, double sign)</u> This method returns the first floating-point argument with the sign of the second floating-point argument.
12	<u>static float copySign(float magnitude, float sign)</u> This method returns the first floating-point argument with the sign of the second floating-point argument.
13	<u>static double cos(double a)</u> This method returns the trigonometric cosine of an angle.
14	<u>static double cosh(double x)</u> This method returns the hyperbolic cosine of a double value.
15	<u>static double exp(double a)</u> This method returns Euler's number e raised to the power of a double value.
16	<u>static double expm1(double x)</u> This method returns $e^x - 1$.
17	<u>static double floor(double a)</u> This method returns the largest (closest to positive infinity) double value that is less than or equal to the argument and is equal to a mathematical integer.
18	<u>static int getExponent(double d)</u> This method returns the unbiased exponent used in the representation of a double.
19	<u>static int getExponent(float f)</u> This method returns the unbiased exponent used in the representation of a float.
20	<u>static double hypot(double x, double y)</u> This method returns $\sqrt{x^2 + y^2}$ without intermediate overflow or underflow.
21	<u>static double IEEERemainder(double f1, double f2)</u> This method returns the natural logarithm (base e) of a double value.
22	<u>static double log(double a)</u> This method returns the natural logarithm (base e) of a double value.
23	<u>static double log10(double a)</u> This method returns the base 10 logarithm of a double value.
24	<u>static double log1p(double x)</u> This method returns the natural logarithm of the sum of the argument and 1.
25	<u>static double max(double a, double b)</u> This method returns the greater of two double values.
26	<u>static float max(float a, float b)</u> This method returns the greater of two float values.
27	<u>static int max(int a, int b)</u>

	This method returns the greater of two int values.
28	<u>static long max(long a, long b)</u> This method returns the greater of two long values.
29	<u>static double min(double a, double b)</u> This method returns the smaller of two double values.
30	<u>static float min(float a, float b)</u> This method returns the smaller of two float values.
31	<u>static int min(int a, int b)</u> This method returns the smaller of two int values.
32	<u>static long min(long a, long b)</u> This method returns the smaller of two long values.
33	<u>static double nextAfter(double start, double direction)</u> This method returns the floating-point number adjacent to the first argument in the direction of the second argument.
34	<u>static float nextAfter(float start, double direction)</u> This method returns the floating-point number adjacent to the first argument in the direction of the second argument.
35	<u>static double nextUp(double d)</u> This method returns the floating-point value adjacent to d in the direction of positive infinity.
36	<u>static float nextUp(float f)</u> This method returns the floating-point value adjacent to f in the direction of positive infinity.
37	<u>static double pow(double a, double b)</u> This method returns the value of the first argument raised to the power of the second argument.
38	<u>static double random()</u> This method returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.
39	<u>static double rint(double a)</u> This method returns the double value that is closest in value to the argument and is equal to a mathematical integer.
40	<u>static long round(double a)</u> This method returns the closest long to the argument.
41	<u>static int round(float a)</u> This method returns the closest int to the argument.
42	<u>static double scalb(double d, int scaleFactor)</u> This method return $d \times 2^{\text{scaleFactor}}$ rounded as if performed by a single correctly rounded floating-point multiply to a member of the double value set.
43	<u>static float scalb(float f, int scaleFactor)</u> This method Return $f \times 2^{\text{scaleFactor}}$ rounded as if performed by a single correctly rounded floating-point multiply to a member of the float value set.
44	<u>static double signum(double d) scaleFactor)</u> This method returns the signum function of the argument; zero if the argument is zero, 1.0 if the argument is

	greater than zero, -1.0 if the argument is less than zero.
45	<u>static float signum(float f)</u> This method returns the signum function of the argument; zero if the argument is zero, 1.0f if the argument is greater than zero, -1.0f if the argument is less than zero.
46	<u>static double sin(double a)</u> This method returns the trigonometric sine of an angle.
47	<u>static double sinh(double x)</u> This method returns the hyperbolic sine of a double value.
48	<u>static double sqrt(double a)</u> This method returns the correctly rounded positive square root of a double value.
49	<u>static double tan(double a)</u> This method returns the trigonometric tangent of an ang
50	<u>static double tanh(double x)</u> This method returns the hyperbolic tangent of a double value.
51	<u>static double toDegrees(double angrad)</u> This method converts an angle measured in radians to an approximately equivalent angle measured in degrees.
52	<u>static double toRadians(double angdeg)</u> This method converts an angle measured in degrees to an approximately equivalent angle measured in radians.
53	<u>static double ulp(double d)</u> This method returns the size of an ulp of the argument.
54	<u>static float ulp(float f)</u> This method returns the size of an ulp of the argument.

Methods inherited

This class inherits methods from the following classes:

- java.lang.Object